InstructionCP: A Simple yet Effective Approach for Transferring Large Language Models to Target Languages

Kuang-Ming Chen[1,2*] **Jenq-Neng Hwang**[1] **Hung-yi Lee**[3]
[1]University of Washington, Seattle, WA, USA
[2]ASUS Open Cloud Infrastructure Software Center, Taipei, Taiwan
[3]National Taiwan University, Taipei, Taiwan
kmchen@uw.edu hwang@uw.edu hungyilee@ntu.edu.tw

Abstract

The rapid development of large language models (LLMs) in recent years has largely focused on English, resulting in models that respond exclusively in English. To adapt these models to other languages, continual pre-training (CP) is often employed, followed by supervised fine-tuning (SFT) to maintain conversational abilities. However, CP and SFT can reduce a model's ability to filter harmful content. We propose Instruction Continual Pre-training (InsCP), which integrates instruction tags—also known as chat templates—into the CP process to prevent loss of conversational proficiency while acquiring new languages. Empirical evaluations on language alignment, reliability, and knowledge benchmarks confirm the efficacy of InsCP. Notably, this approach requires only 0.1 billion tokens of high-quality instruction-following data, thereby reducing resource consumption.

1 Introduction

Large language models (LLMs) have demonstrated remarkable performance across numerous natural language processing (NLP) tasks(Brown et al., 2020). However, the majority of LLMs are pretrained on English corpora(AI@Meta, 2024; Team et al., 2024; OpenAI, 2023), thus restricting their utility to English language contexts.

While some endeavors opt to train their LLMs from scratch using non-English data, as exemplified by YI-34B(AI et al., 2024), we recognize the significant time and computing resources required for such an approach. Drawing inspiration from Ouyang et al. (2022), many research groups have shifted their focus towards continual pre-training (CP)(Gupta et al., 2023; Ke et al., 2022) on target languages to enhance knowledge acquisition and model fluency. Subsequently, supervised fine-tuning (SFT) is conducted on instruction-formatted data to ensure that models possess the capability to respond to questions in a format consistent with English-based pretrained LLMs, such as BLOOM(Workshop et al., 2023), LLaMA2(Touvron et al., 2023), and Mistral-7B(Jiang et al., 2023).

Yet, as highlighted in Qi et al. (2023), challenges persist in maintaining RLHF capabilities when fine-tuning GPT-3.5 turbo(OpenAI, 2023) on non-English data. Our experiments validate similar observations with other LLMs like LLaMA2.

This work proposes a novel fine-tuning approach called Instruction Continual Pre-training (InsCP) for LLMs to adapt to non-English languages. We hypothesize that providing a chat template during CP prevents the model from forgetting its conversational abilities, as it mirrors its original training conditions. InsCP is essentially the same as typical CP, except that we augment each piece of data with a chat template containing special instruction tokens, such as $< |begin\_of\_text| >$ in LLaMA3(AI@Meta, 2024). This simple augmentation enables the model to effectively retain its original RLHF capabilities, such as defending against offensive input while learning a new language through CP.

We evaluate the effectiveness of InsCP on LLMs, primarily focusing on the LLaMA3-instruct model, across three key aspects: language alignment, reliability, and knowledge benchmarks.

The results demonstrate that the model, after undergoing InsCP on LLaMA3-instruct, effectively performs in Traditional Chinese when prompted with Traditional Chinese input, surpassing the performance of LLaMA3-instruct. Moreover, the model after InsCP does not suffer a serious performance dropped in knowledge, safety and RLHF ability.

[*] This work was conducted while the first author was an intern at ASUS Open Cloud Infrastructure Software Center.

Proceedings of the 7th Workshop on Research in Computational Linguistic Typology and Multilingual NLP (SIGTYP 2025), pages 1–6
August 1, 2025 ©2025 Association for Computational Linguistics

2 Related Work

2.1 LLMs adapt in other languages

Fine-tuning is a widely-used technique for adapting models, particularly in the domain of large language models (LLMs), to specific domains. Many downstream tasks have been successfully addressed through fine-tuning (Howard and Ruder, 2018; Devlin et al., 2019; Radford et al., 2018). While most downstream tasks can be accomplished through supervised fine-tuning, adapting an English-based LLM to other languages, such as in the work of Fujii et al. (2024); Zhao et al. (2024); Cui et al. (2023); Lin and Chen (2023); YuLan-Team (2023) for non-English languages, typically begins with continual pre-training (CP). This initial step is crucial for ensuring that the models possess the necessary language proficiency and knowledge. Since acquiring proficiency in a specific language requires a large amount of data, CP is advantageous as it does not require labeled data, enabling the use of vast amounts of available language data. Subsequently, instruction fine-tuning allows the model to engage in conversational interactions using specific templates.

2.2 Problems of Continual Pre-training

Continual pre-training (CP) is often employed to adapt those English-based models to other languages. However, Li and Lee (2024) points out that CP can lead to catastrophic forgetting, particularly diminishing the model's conversational abilities. To address this issue, Huang et al. (2024) proposed a method called "chat vector," which enhances chat capabilities through model weight arithmetic, achieving good performance across various benchmarks. Despite these advancements, many researchers continue to tackle the challenges posed by CP. In this work, we present a straightforward approach to mitigate these issues.

3 Methodology

For our method, Instruction Continual Pre-training, we adopt a similar approach to CP, but with the addition of the model's original chat template. The template is shown in Appendix A.1 The **inputs** in the template represent the prompts provided by the user. In our context, where the objective is to train LLMs in the target language through next token prediction tasks while retaining their chat ability, we place the CP data in the **model_response**. This arrangement ensures that LLMs generate tokens based on the target language. The InsCP template is shown in A.1.

4 Experimental Setup

4.1 Pre-training Dataset

We utilize a high-quality dataset comprising paired instruction-following data for LLaMA3-instruct 8B(AI@Meta, 2024) during the InsCP procedure. The InsCP procedure means the traditional CP method with instruction-following data. The dataset consists of Traditional Chinese text and has a total size of 0.1 billion tokens. Throughout the InsCP process, we segregate the questions and answers into two separate data points. Further details regarding the training process are provided in the Appendix A.3.

Moreover, to demonstrate the generalizability of our method to other languages, we extend our approach to Japanese. We utilize a 70M tokens dataset, which is also instruction-following data same as the Traditional Chinese dataset structure, to perform InsCP on LLaMA3-instruct 8B.

From our experiments, we discovered the critical importance of selecting appropriate data for InsCP. We aimed to determine the most suitable type of data for InsCP. Based on our findings, we selected instruction-following data with low perplexity because low perplexity are likely to closely resemble the original output of LLMs, thereby minimizing any adverse effects on the models' original abilities.

4.2 Evaluation

4.2.1 Language Alignment

To evaluate language alignment, we employ the FastText language identification model (Joulin et al., 2016a,b). This model is used to determine the language of 2000 aligned sentences extracted from the English and Traditional Chinese subset of the NeuLab-TedTalks language within the tokens generated by our model. The FastText model classifies text into two categories: Chinese and English. The results include the percentage of sentences identified as Chinese, English, and others from the set of 2000 input prompts.

4.2.2 Reliability

We assess the reliability of the model's output using several common benchmarks, including TruthfulQA(Lin et al., 2022), ToxiGen(Hartvigsen et al., 2022), and BOLD(Dhamala et al., 2021), utilizing lm-evaluation-harness(Gao et al., 2021).

4.2.3 Knowledge Benchmarks

We utilize several benchmarks to evaluate our model's knowledge: **C-eval-tw**: A translation of C-eval(Huang et al., 2023), used to evaluate our model. Compute metrics by averaging accuracy across individual tasks. The accuracy computation involves selecting the option with the highest probabilities. **TTQA**(Hsu et al., 2023): Focuses on Taiwanese commonsense and knowledge by using 64 expert-selected paragraphs from Wikipedia. We extract the model's output and calculate accuracy based on multiple-choice questions. **TMMLU Plus**(Tam et al., 2024): Used for traditional Chinese multitask benchmarking. We calculate accuracy for each task directly. **ARC**(Clark et al., 2018) and **Hellaswag**(Zellers et al., 2019): Ensure that our model's English-related knowledge does not degrade. We utilize length-normalized accuracy. **MMLU**(Hendrycks et al., 2020): Suitable for multitask evaluation. We calculate accuracy for each task directly.

4.2.4 MT-Bench

MT-Bench(Zheng et al., 2023) incorporates multi-conversation scenarios, allowing us to assess the model's ability to handle multiple interactions simultaneously. This enables us to demonstrate that InsCP does not compromise the RLHF ability of the model. In MT-Bench, the GPT-4 score serves as our evaluation metric, and we include a prompt about judging language alignment in GPT-4 evaluation to test the model's language ability.

4.3 Baselines

We select LLaMA-3-instruct as our baseline model. To evaluate the performance of Instruction Continual Pre-training (InsCP), we conduct InsCP using our baseline model. Importantly, it's worth noting that both InsCP and the original continual pre-training (orgCP) utilize the same continual pre-training (CP) data. Furthermore, to compare with the original continual pre-training process, we also fine-tune a model using original continual pre-training.

Model	**EN Prompt**		**ZH Prompt**	
	EN% ↑	**ZH% ↓**	**EN% ↓**	**ZH% ↑**
LLaMA3-instruct	1.0	0.0	0.90	0.09
LLaMA3-orgCP	1.0	0.0	0.50	0.49
LLaMA3-InsCP	0.99	0.01	0.01	**0.99**

Table 1: Language alignment benchmark.

model	TruthfulQA mc2 ↑		ToxiGen toxicity ↓		BOLD sentiment ↓	
language	EN	ZH	EN	ZH	EN	ZH
LLaMA3-instruct	51.6	52.7	0.10	0.14	0.54	0.55
LLaMA3-orgCP	50.8	50.5	0.12	0.26	0.61	0.68
LLaMA3-InsCP	**51.8**	**53.8**	**0.07**	0.16	0.56	**0.52**

Table 2: Reliability benchmark

5 Experimental Result

5.1 Language alignment evaluation

We present the percentage of responses among 2000 prompts generated by the models. The experimental findings are summarized in Table 1. Our observations are as follows: (1)**LLaMA3-instruct exhibits poor language alignment:** As indicated in Table 1, when provided with Traditional Chinese input prompts, LLaMA3-instruct frequently generates output in English. This lack of alignment between the input and output languages can lead to language nonalignment issues during usage. (2)**The same data used with the original CP method fails to achieve proper alignment:** A key distinction between InsCP and the original CP lies in their respective language learning capabilities. We observed that with the same data size, InsCP enables LLMs to acquire language proficiency more effectively. (3)**LLaMA3-InsCP demonstrates remarkable language proficiency:** Regardless of whether provided with English or Traditional Chinese input prompts, LLaMA3-InsCP consistently responds in the appropriate language.

5.2 Reliability evaluation

In Table 2, we present the results of the models' reliability. Our experiments were conducted in both English and Chinese to ensure that our model does not compromise its RLHF ability in either language. Across each benchmark, we observe that the orgCP model consistently achieves lower scores compared to the other models. On the other hand, LLaMA3-InsCP retain the RLHF ability, allowing it to defend against toxic inputs and generate nonharmful context during inference.

5.3 Knowledge benchmark

In Table 3, we present the scores from six knowledge benchmark tests. In Chinese-related benchmarks, we observed that the model after InsCP exhibited some improvements compared to both orgCP and the original model. These findings indicate that InsCP can effectively preserve the LLM's

model	ARC	Hellaswag	MMLU	C-eval-tw	TMMLU+	TTQA
	ACC ↑	ACC ↑	ACC ↑	ACC ↑	ACC ↑	ACC ↑
LLaMA3-instruct	60.5	81.8	67.2	47.3	43.0	23.3
LLaMA3-orgCP	57.5	81.3	66.1	48.5	41.3	41.3
LLaMA3-InsCP	**61.6**	81.7	65.6	**48.9**	**41.9**	**48.5**

Table 3: Knowledge benchmark

model	MT-Bench	
language	EN ↑	ZH ↑
LLaMA3-instruct	7.8	4.1
LLaMA3-orgCP	4.3	4.6
LLaMA3-InsCP	7.6	**6.7**

Table 4: MT-Bench

model	MT-Bench-JP
LLaMA3-instruct	4.9
LLaMA3-orgCP-JP	4.8
LLaMA3-InsCP-JP	6.6

Table 5: MT-Bench-JP

inherent abilities while also enhancing its performance in target language domains.

5.4 MT-Bench and MT-Bench-JP

In Tables 4 and 5, MT-Bench further highlights the distinctions between orgCP and InsCP. We note that outputs from orgCP often contain irrelevant text that deviates from our input prompts. Moreover, the orgCP model appears to forget how to appropriately conclude conversations. Additionally, due to the inclusion of language alignment criteria in GPT-4 evaluation, we observe a significant disparity between the InsCP model and LLaMA3-instruct. While LLaMA3-instruct predominantly responds in English for most questions, the InsCP model demonstrates the ability to discern the language input by the user. We observe a distribution similar to that of Traditional Chinese MT-Bench in Table 5 in Japanese domain.

6 Limitations of InsCP

As discussed in Section 4.1, the choice of data used in InsCP significantly influences its outcomes. Our experiments indicate that conducting InsCP necessitates the utilization of low-perplexity instruction-following data, which can be challenging to acquire in abundance for certain languages. Consequently, we opted to perform InsCP using small datasets, which we believe is a more generalizable approach for languages with limited resources. Nonetheless, both data size and data quality remain challenges when implementing InsCP.

7 Conclusion

In this work, we introduce a novel pipeline called InsCP designed to facilitate the transfer of LLMs into non-English domains. Through InsCP, LLMs can retain their inherent abilities while also acquiring the capability for language alignment in the target language and gaining knowledge of the target domain. Additionally, we demonstrate that InsCP does not necessitate extensive data, thereby consuming fewer resources and less time. Remarkably, even with a small amount of data, InsCP can transform English-based LLMs into models aligned with the target language, a stark contrast to the resource-intensive traditional pipeline. InsCP paves the way for future LLMs, primarily finetuned in specific languages, to swiftly transfer their abilities to other languages.

References

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai.

AI@Meta. 2024. Llama 3 model card.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and Rahul Gupta. 2021. Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21. ACM.

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual pre-training for cross-lingual llm adaptation: Enhancing japanese language capabilities.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats Leon Richter, Quentin Gregory Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. 2023. Continual pre-training of large language models: How to re-warm your model? In *Workshop on Efficient Systems for Foundation Models @ ICML2023*.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification.

Chan-Jan Hsu, Chang-Le Liu, Feng-Ting Liao, Po-Chun Hsu, Yi-Chang Chen, and Da shan Shiu. 2023. Advancing the evaluation of traditional chinese language models: Towards a comprehensive benchmark suite.

Shih-Cheng Huang, Pin-Zu Li, Yu-Chi Hsu, Kuang-Ming Chen, Yu Tung Lin, Shih-Kai Hsiao, Richard Tzong-Han Tsai, and Hung yi Lee. 2024. Chat vector: A simple approach to equip llms with instruction following and model alignment in new languages.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2022. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations*.

Chen-An Li and Hung-Yi Lee. 2024. Examining forgetting in continual pre-training of aligned large language models.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods.

Yen-Ting Lin and Yun-Nung Chen. 2023. Language models for taiwanese culture. Code and models available at https://github.com/MiuLab/Taiwan-LLaMa.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2023. Fine-tuning aligned language models compromises safety, even when users do not intend to!

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Zhi-Rui Tam, Ya-Ting Pai, Yen-Wei Lee, Sega Cheng, and Hong-Han Shuai. 2024. An improved traditional chinese evaluation suite for foundation model.

Gemini Team, Rohan Anil, Sebastian Borgeaud, and Jean-Baptiste Alayrac et al. 2024. Gemini: A family of highly capable multimodal models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, and Matthias Gallé *et al.* 2023. Bloom: A 176b-parameter open-access multilingual language model.

YuLan-Team. 2023. Yulan-chat: An open-source bilingual chatbot. `https://github.com/RUC-GSAI/YuLan-Chat`.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence?

Jun Zhao, Zhihao Zhang, Luhui Gao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llama beyond english: An empirical study on language capability transfer.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

A Appendix

A.1 LLaMA3-instruct chat template

To initiate a completion with LLaMA3-instruct, one must adhere to the following format:

```
<| begin_of_text |>
<| start_header_id |>user<|end_header_id|>
{{ inputs }}<| eot_id |>

<| start_header_id |> assistant <|end_header_id
    |>
{{model_response}}
```

The InsCP template is shown below:

```
<| begin_of_text |>
<| start_header_id |>user<|end_header_id|><|
    eot_id |>
<| start_header_id |> assistant <|end_header_id
    |>
{{InsCP_data}<|eot_id|>}
```

A.2 Training Detail

We utilize LLaMA3-instruct as our base model, and both the original continual pre-training and instruction continual pre-training are configured with the following hyperparameters: a learning rate of 3e-5, AdamW optimizer with beta1 of 0.9 and beta2 of 0.95, batch size set to 1 per device (utilizing 64 GPUs), and training conducted for 10 epochs.

A.3 Generation Strategy

We employ vLLM as our generation tool, incorporating LLaMA3's system prompt in each generation to harness the full potential of the LLM. For vLLM, we set the following generation parameters: maximum tokens to 1024, temperature to 0.8, top-p sampling to 0.9, and seed fixed at 42 to facilitate result reproducibility. Additionally, we maintain default values for other generation configurations in vLLM.

A.4 MT-Bench evaluation prompt

In the Traditional Chinese MT-Bench, we predominantly adhere to the evaluation prompts provided by the authors. However, to delve deeper into testing the LLM's language alignment ability, we introduce an additional prompt in Traditional Chinese: "If the assistant's answer is in a language other than Traditional Chinese, please give it a score of 0." This prompt instructs GPT-4 to assign a score of 0 to responses that are not in the correct language, thereby enabling a more rigorous assessment of language alignment capabilities. For Japanese MT-Bench, we also add the prompt in Japanese: "If the assistant's answer is in a language other than Japanese, please give it a score of 0.", in order to meet the language alignment requirement we want to obseve.